



Processamento Digital de Sinais

Alunos:

Gabriel Victor M. de O. Vital

Pablo Godoy

Professor:

Ricardo Campello

Universidade federal de
Pernambuco
Centro de Tecnologia e Geociências
Departamento de Eletrônica e Sistemas

Detecção de sinais DTMF

Relatório referente ao projeto da terceira unidade.

**Alunos: Gabriel Victor Marques
Pablo Godoy
Professor: Ricardo Campello**

Outubro, 2020

Conteúdo

1	Introdução	1
2	Objetivo	1
3	Parte 1: Problemas Básicos	3
3.1	Parte 1.1 - (a)	3
3.2	Parte 1.2 - (b)	3
3.3	Parte 1.3 - (b)	3
4	Parte 2: Problemas Intermediários	4
4.1	Parte 2.1 - (d)	4
4.2	Parte 2.1 - (e)	13
5	Parte 3: Problemas Avançados	18
5.1	Parte 3.1 - (f)	18
5.2	Parte 3.2 - (g)	18
5.3	Parte 3.2 - (h)	19
6	Conclusão	21
7	Apendice	22
	Bibliografia	26

1 Introdução

Os sinais DTFM (Dual-TOne Multifrequency) tem diversas aplicações importantes, sendo usada em muitos sistemas de Telecomunicações, telefonia a teclado, correio de voz, sistemas bancários interativos e comércio eletrônico são exemplos de suas aplicações. A crescente demanda por taxas de transmissão cada vez mais elevadas, faz com que a velocidade de decodificação desses sinais(DTMF) tornem-se um parâmetro importante no projeto desses sistemas. [4]

2 Objetivo

Neste projeto terá como objetivo demonstrar como o sistema de teclas em um telefone usa sinais de diferentes frequências para indicar qual tecla foi pressionada. A Transformada de Fourier de Tempo Discreto (TFTD) de um sinal de telefone amostrado pode ser usada para identificar essas frequências. O som que você escuta quando uma tecla é pressionada é a soma de duas senóides. A senóide de frequência mais alta indica a coluna e a de frequência mais baixa indica a linha do teclado. A figura 1 abaixo mostra um teclado juntamente com as duas frequências correspondentes a cada dígito, considerando que a forma de onda contínua é amostrada a 8192 kHz. A figura mostra ainda uma tabela contendo as frequências de cada dígito. Assim, por exemplo, o dígito 5 é representado pela equação 2.0.1 . [3]

$$d_5 = \text{sen}(0,5906n) + \text{sen}(1,0247n) \quad (2.0.1)$$

o caso geral para as frequências w_i e w_j listadas na tabela da figura 1 temos que:

$$d_k = \text{sen}(w_i n) + \text{sen}(w_j n) \quad (2.0.2)$$

em que k é o dígito, w_i se refere a frequência da linha e w_j é referente a frequência da coluna.

	Freq. (colunas)		
Freq. (linhas)	0,9273	1,0247	1,1328
0,5346	1	2	3
0,5906	4	5	6
0,6535	7	8	9
0,7217		0	

(a)

Digito	Frequência (Linha)	Frequência (Coluna)
0	0,7217	1,0247
1	0,5346	0,9273
2	0,5346	1,0247
3	0,5346	1,1328
4	0,5906	0,9273
5	0,5906	1,0247
6	0,5906	1,1328
7	0,6535	0,9273
8	0,6535	1,0247
9	0,6535	1,1328

Figura 1: Frequências da DTFT para os tons de um telefone a teclado.

Neste projeto será mostrado como o sistema de teclas em um telefone usa sinais de diferentes frequências para indicar qual tecla foi pressionada. A Transformada de Fourier de Tempo Discreto (TFTD) de um sinal de telefone amostrado pode ser usada para identificar essas frequências. O som que você escuta quando uma tecla é pressionada é a soma de duas senóides. A senóide de frequência mais alta indica a coluna e a de frequência mais baixa indica a linha do teclado.^[3]

3 Parte 1: Problemas Básicos

Nesses problemas será criado tons apropriados para cada dígito e examinaremos a TFTD para verificar se os sinais tem as frequências corretas. Também definiremos um vetor contendo os tons de seu número de telefone.

3.1 Parte 1.1 - (a)

Nesta parte foi implementado um script no software MATLAB para criar um vetor de linha de d0 a d9 aos quais representam 10 dígitos num intervalo $0 \leq n \leq 999$. Cada sinal foi escutado por meio da função **sound**, foi notado que os sinais são bem definidos e que é bastante claro ao analisar a equação 2.0.2 da figura 1.

3.2 Parte 1.2 - (b)

Nesta parte foi utilizado a função **fft** para computar N amostras da TFTD de um sinal de comprimento finito nas frequências de $\omega_k = \frac{2k\pi}{N}$, essas amostras são computadas igualmente espaçadas. A **fft** foi utilizada em específico para computar amostras de $D_2(e^{j\omega})$ e $D_9(e^{j\omega})$ em $\omega_k = \frac{2k\pi}{2048}$. um vetor foi criado contendo os ω_k para $0 \leq k \leq 2047$, a amplitude da TFTD para esses sinais foram obtidas e está disposta na figura 2.

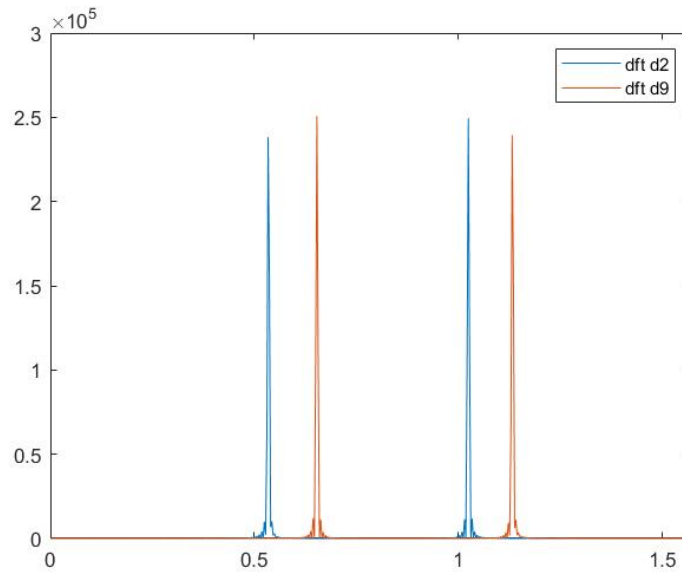


Figura 2: TFTD dos dígitos d_2 e d_9

3.3 Parte 1.3 - (b)

Nesta parte a função foi criado um vetor com 100 amostras com valor zero para representar o "silêncio". Foi definido um número de telefone como **fone**

utilizando os sinais corretos que tem suas frequências dispostas na tabela da figura 1. Como exemplo, se o número é 32718210, vai ser digitado:

```
fone = [d3 space d2 space d7 space d1 space d8 space d2 space d1 space d0].
```

Por fim, foi utilizado-se a função **sound** para tocar o número digitado (fone).

4 Parte 2: Problemas Intermediários

Nesta parte o desafio será decodificar números de telefone que estão dispostos no arquivo **touch.mat**. Os dados foram carregados no MATLAB utilizando o comando **load touch**, após o arquivo ser carregado corretamente fomos capazes de listar os nomes das variáveis digitando o comando **who**, as variáveis são:

```
hardx1 hardx2 x1 x2
```

Os vetores **x1** e **x2** contém versões amostradas das sequências representando dois diferentes números (as sequências são como na parte (c)). Os vetores **hardx1** e **hardx2** são versões discadas menos precisas dos mesmos números.

4.1 Parte 2.1 - (d)

Na parte (d) utilizamos o comando **fft** para computar as 2048 amostras igualmente espaçadas da TFTD de cada dígito de **x1**. As TFTDs dos dígitos de **x1** estão dispostas nas figuras 3, 4, 5, 6, 7, 8 e 9.

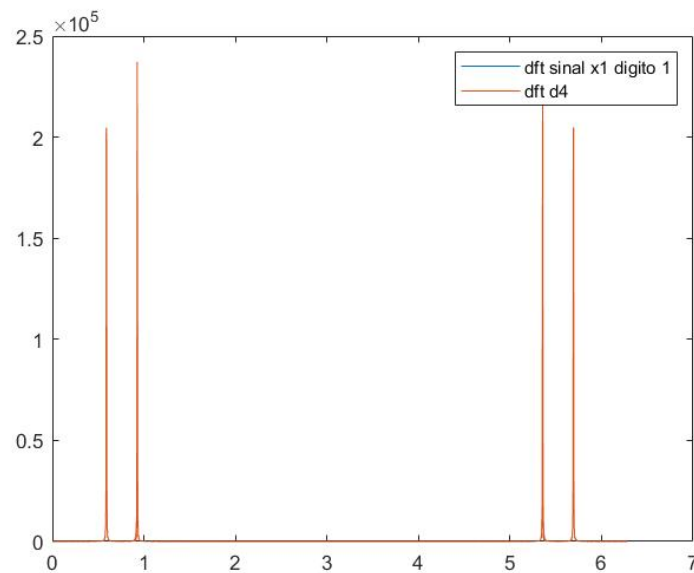


Figura 3: Magnitude da TFTD do primeiro dígito de **x1**

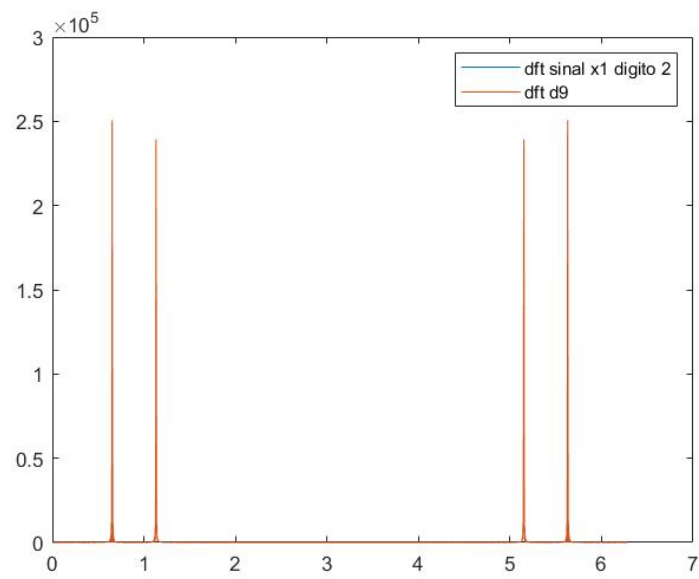


Figura 4: Magnitude da TFTD do segundo dígito de x1

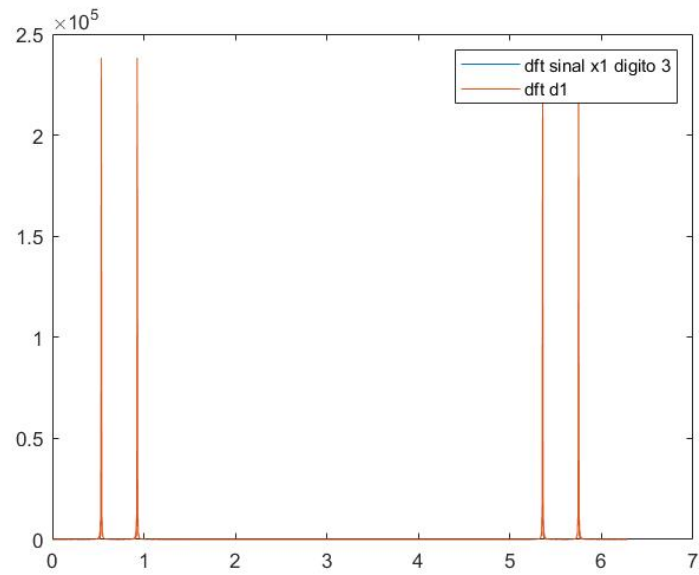


Figura 5: Magnitude da TFTD do terceiro dígito de x1

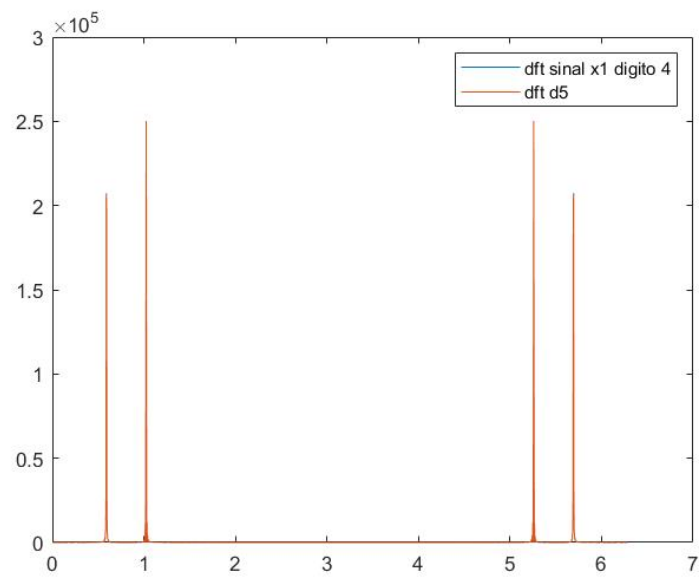


Figura 6: Magnitude da TFTD do quarto dígito de x1

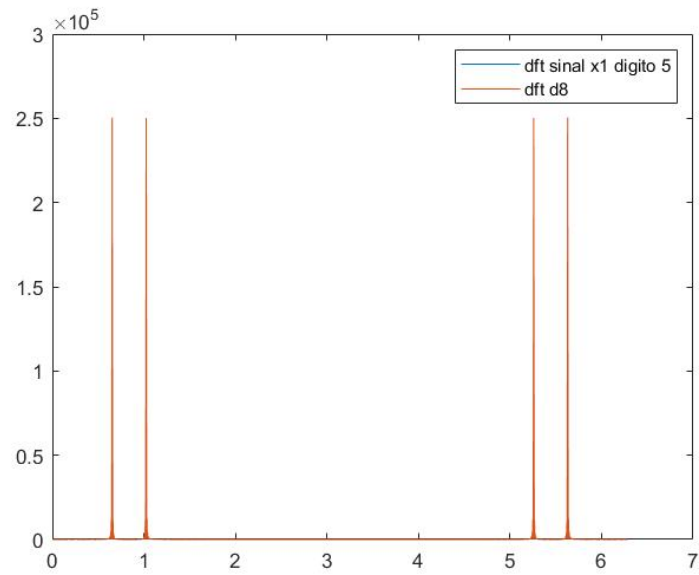


Figura 7: Magnitude da TFTD do quinto dígito de x1

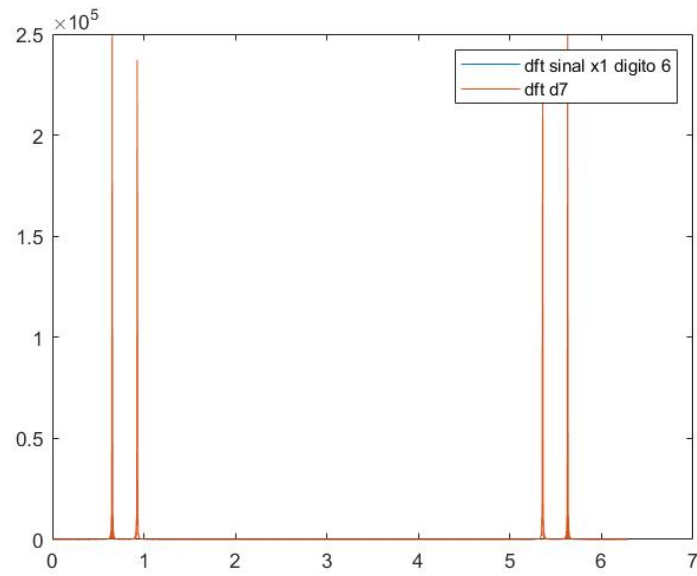


Figura 8: Magnitude da TFTD do sexto dígito de x_1

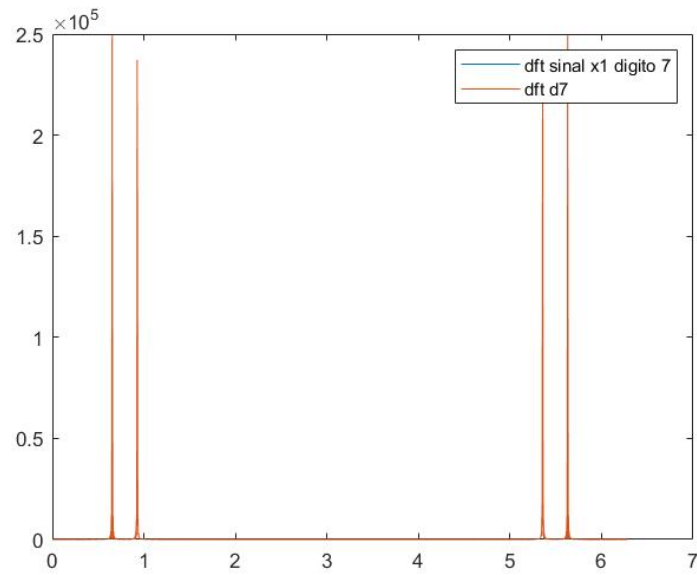


Figura 9: Magnitude da TFTD do sétimo dígito de x_1

As TFTDs dos dígitos **0**, **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**, **9**, cujas frequências de linhas e colunas estão dispostas na parte (b) da figura 1, podem ser observadas

nas figuras [10](#), [11](#), [12](#), [13](#), [14](#), [15](#), [16](#), [17](#), [18](#) e [19](#).

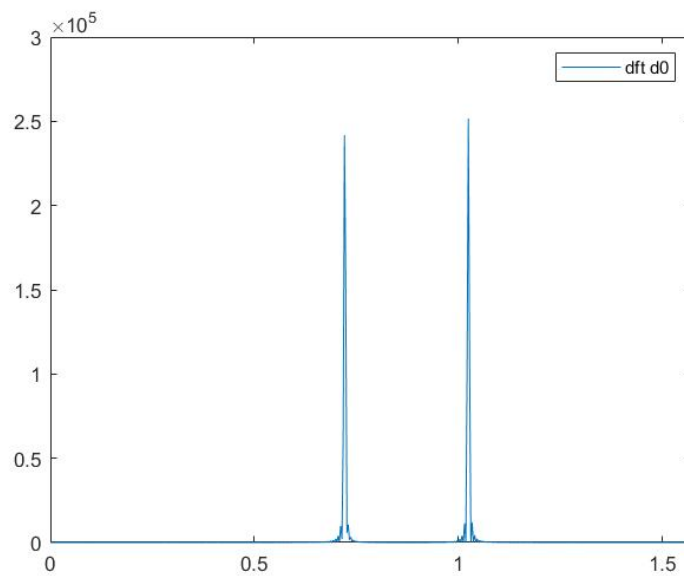


Figura 10: TFTD do dígito 0

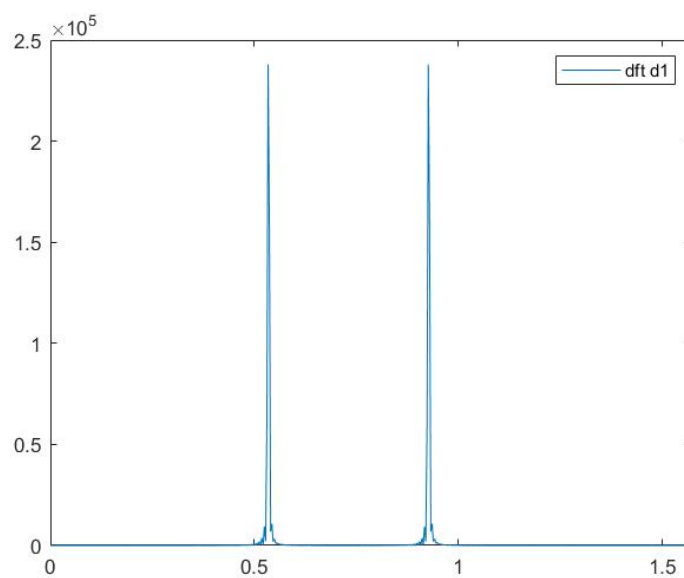


Figura 11: TFTD do dígito 1

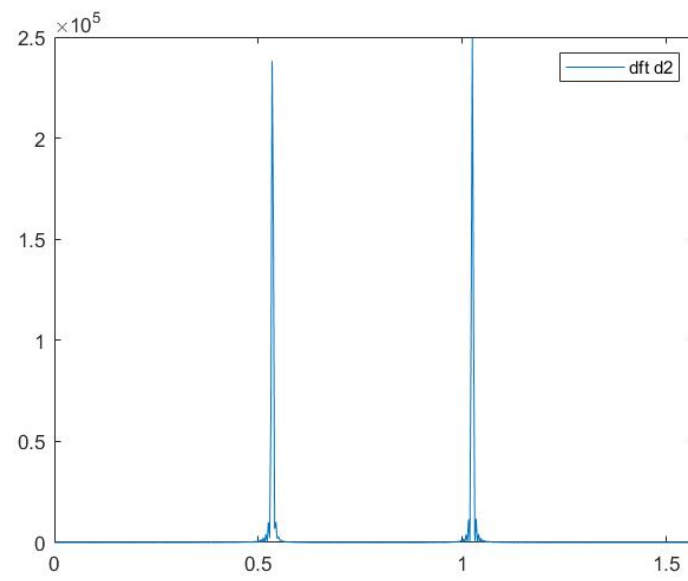


Figura 12: TFTD do dígito 2

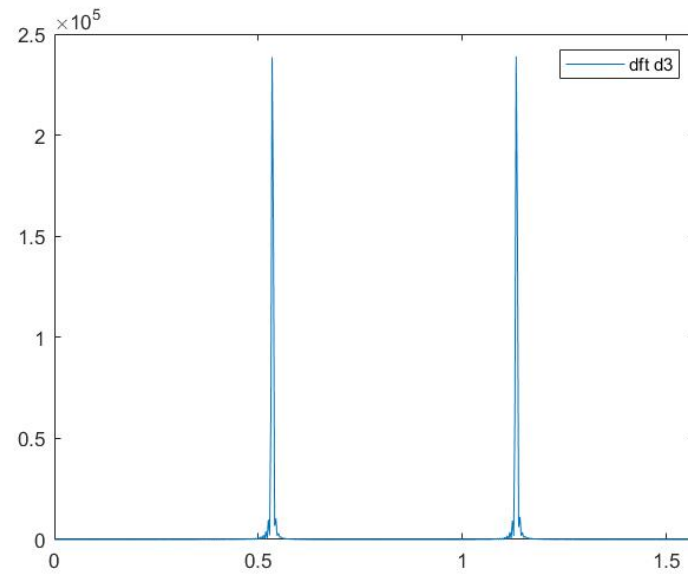


Figura 13: TFTD do dígito 3

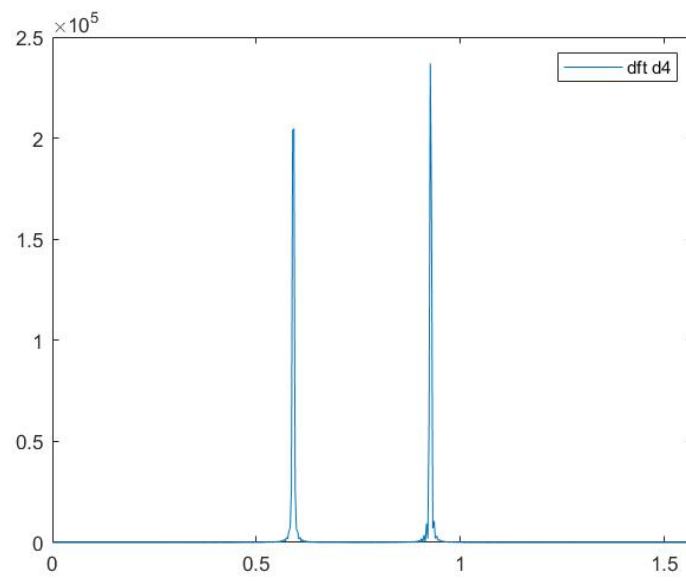


Figura 14: TFTD do dígito 4

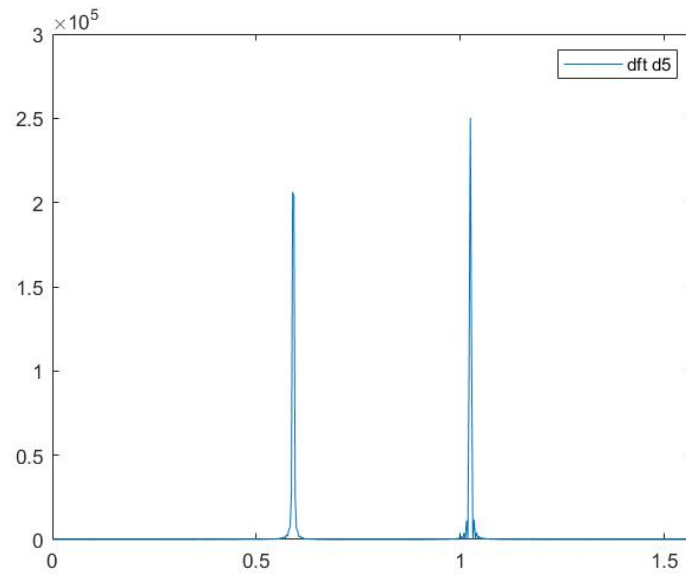


Figura 15: TFTD do dígito 5

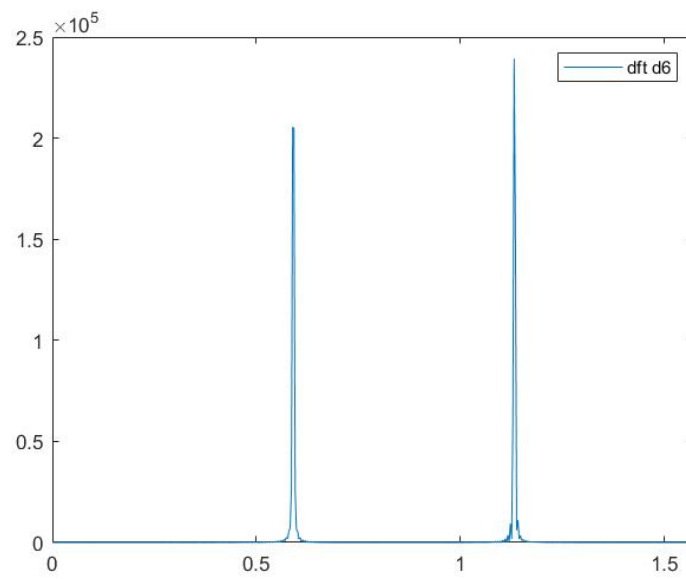


Figura 16: TFTD do dígito 6

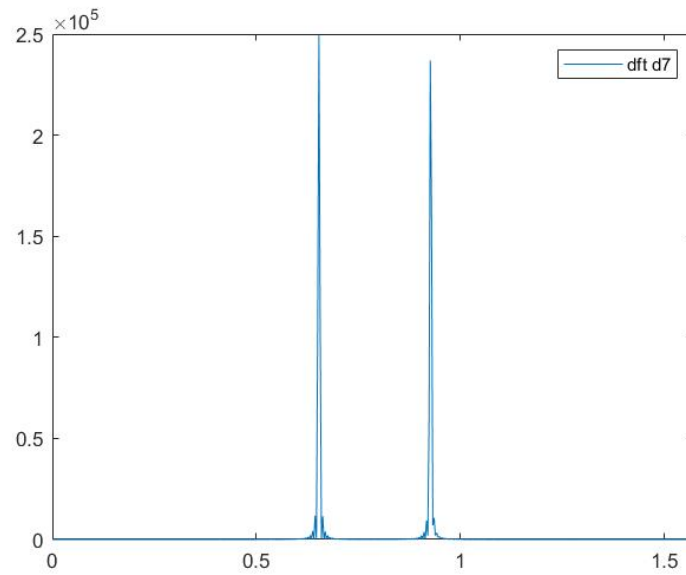


Figura 17: TFTD do dígito 7

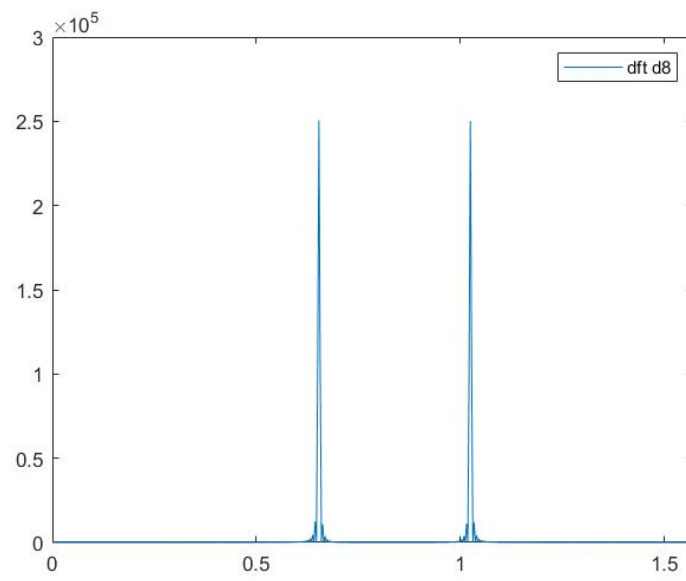


Figura 18: TFTD do dígito 8

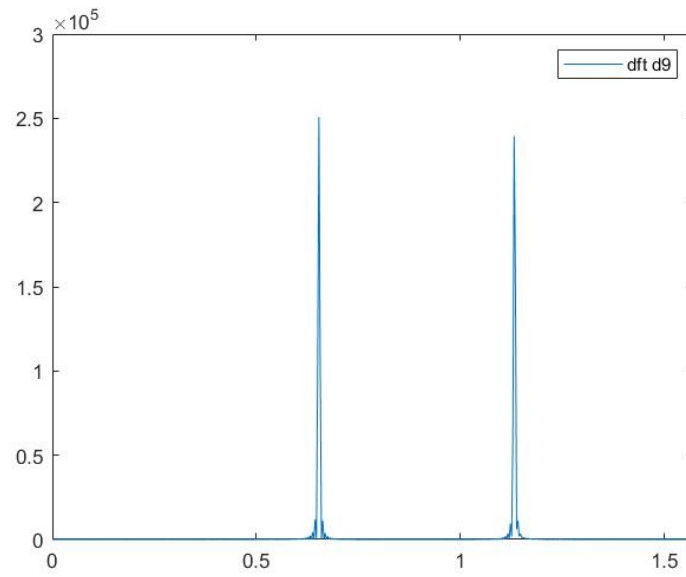


Figura 19: TFTD do dígito 9

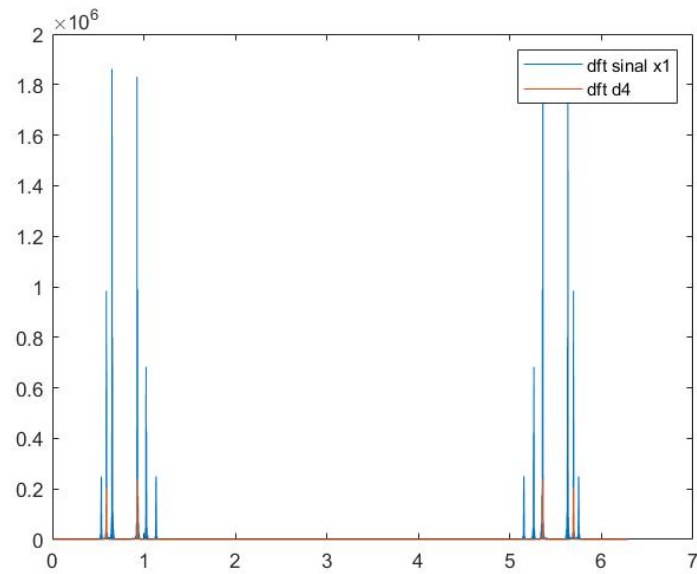


Figura 20: TFTD dos dígitos de x1

Ao comparar as TFTDs dos dígitos de x1 com as TFTDs dos dígitos de 0 à 9 foi possível determinar os dígitos do número de telefone que estão codificado em x1. E ao comparar as frequências de pico do sinal com aquelas da Figura 20 vimos que a soma dos dígitos do número encontrado foi 41. O número encontrado foi é **4 9 1 5 8 7 7**.

4.2 Parte 2.1 - (e)

Foi repetido o procedimento da parte (d) para o sinal x2. As TFTDs dos dígitos de x2 estão dispostas nas figuras 21, 22, 23, 24, 25, 26 e 27.

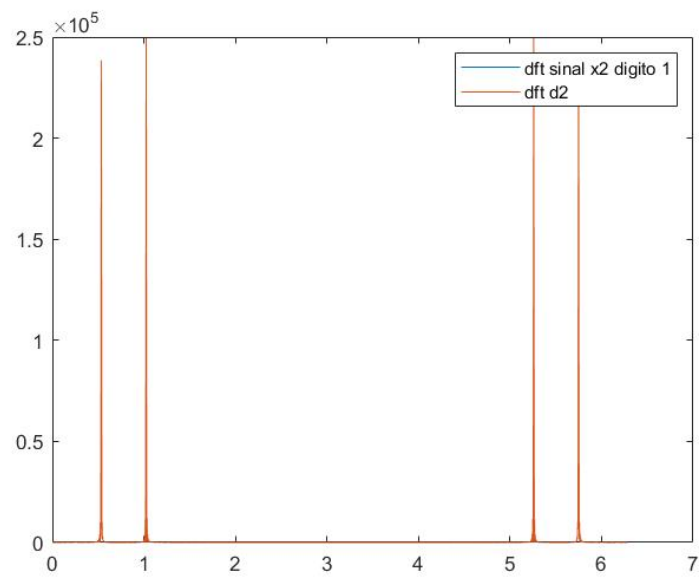


Figura 21: TFTD do primeiro dígito de x_2

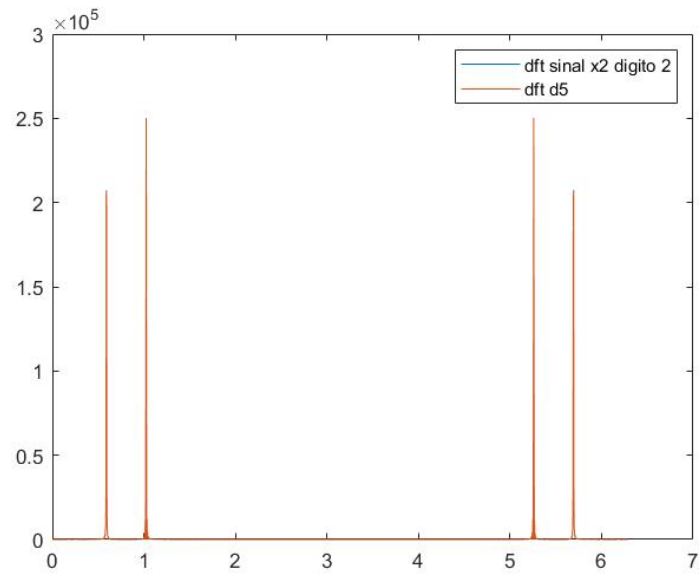


Figura 22: TFTD do segundo dígito de x_2

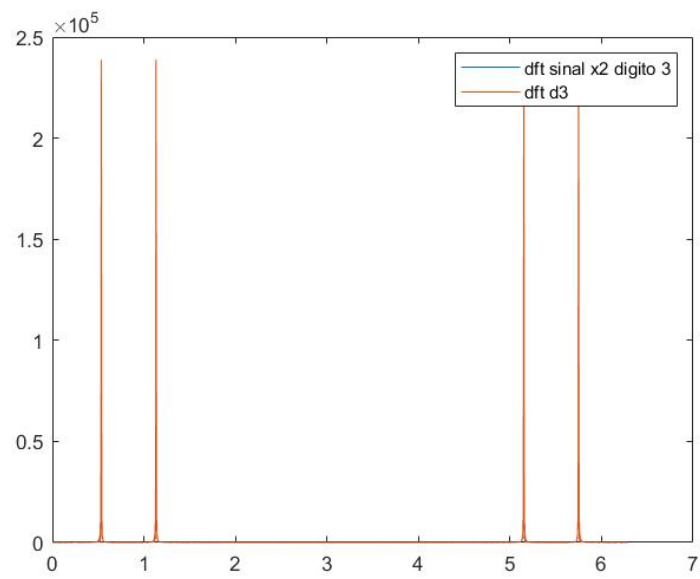


Figura 23: TFTD do terceiro dígito de x_2

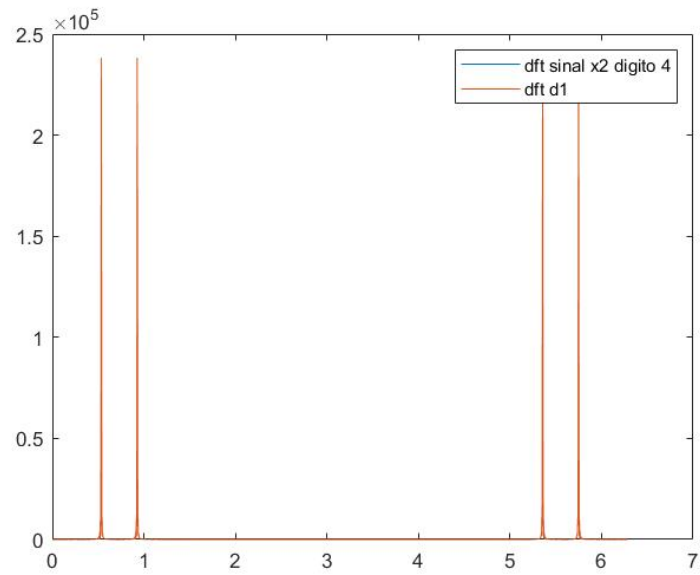


Figura 24: TFTD do quarto dígito de x_2

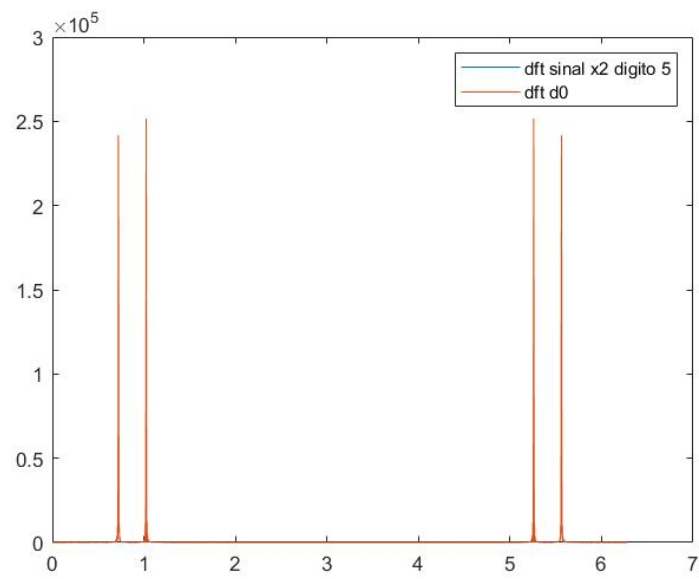


Figura 25: TFTD do quinto dígito de x_2

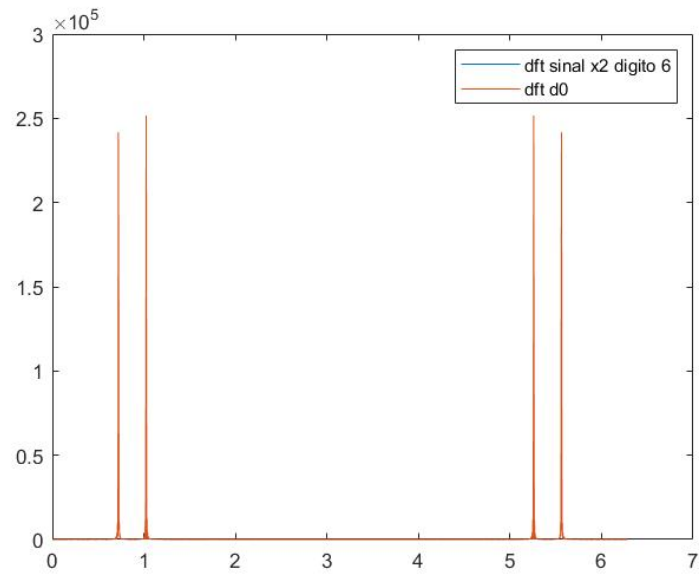


Figura 26: TFTD do sexto dígito de x_2

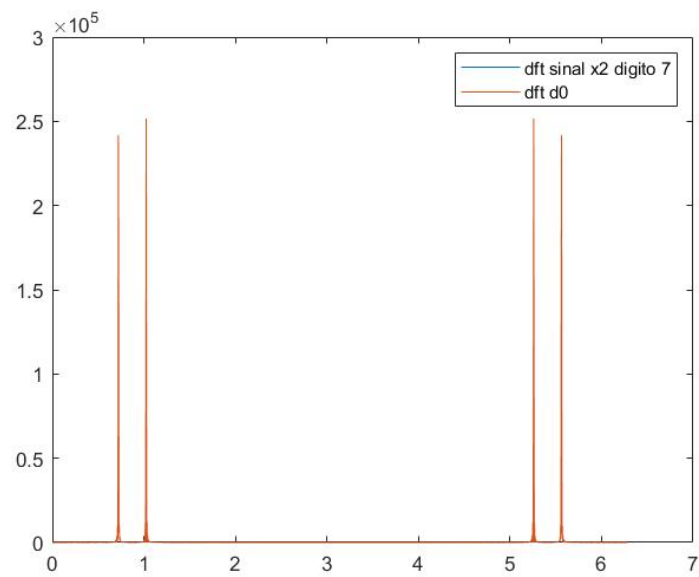


Figura 27: TFTD do sétimo dígito de x2

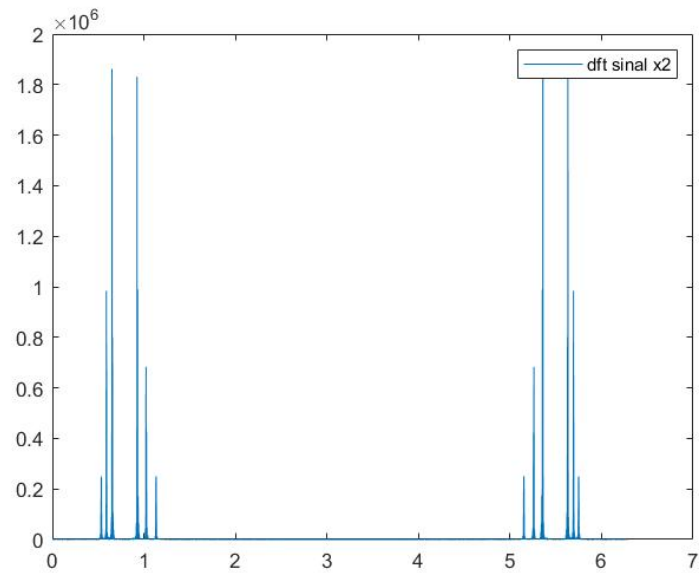


Figura 28: TFTD dos dígitos de x2

A Figura 28 mostra a TFTD de x2 e ao comparar as frequências de pico do sinal com de cada dígito ,individualmente, de x2 com as TFTDs dos dígitos

de **0** à **9** o número telefônico encontrado foi **2 5 3 1 0 0 0** e a soma dos seus dígitos é 11.

5 Parte 3: Problemas Avançados

Na seção de problemas avançados escrevemos uma função para decodificar números de telefone automaticamente a partir dos tons gerados por um teclado. Para projetar o decodificador foi observado as energias dos tons em cada uma das possíveis frequências indicadas na figura 1.

5.1 Parte 3.1 - (f)

Nesta parte usamos a fft para computar 2048 amostras de $X(e^{jw})$ e foi encontrado o valor de w_k e do índice k correspondente, em que esses w'_k s são as frequências dos sons e o resultado pode ser visualizado na na Tabela 1.

Tabela 1: Tabela de valores do índice k e W_k

k	w_k
175	0.5338
193	0.59212
214	0.65348
236	0.72097
303	0.92652
335	1.02470
370	1.13208

5.2 Parte 3.2 - (g)

O valor de $|X(e^{jw_k})|^2$ da a energia em um sinal na frequência w_k . Com isso foi calculado a DFT de d_8 que foi definido como na parte 1 - (a). A saída da fft foi usada para computar $|D_8(e^{jw})|^2$ para cada valor de w_k que foram determinados em (f). O grafico da energia do sinal está disposta na figura 29.

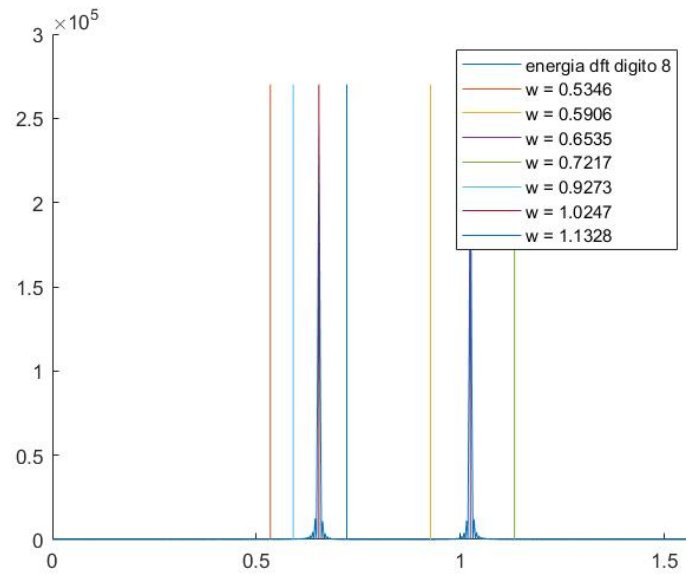


Figura 29: Energia de d_8

5.3 Parte 3.2 - (h)

Nesta parte foi escrita o script de uma função que foi denominada de **dectmf**, o script está disposto no apêndice, este tem como entrada um sinal DTMF no formato usado na parte (c) e que retorna na saída um vetor de comprimento 7 contendo o número de telefone. A função foi testada utilizando x_1 e x_2 como entrada e foi verificado que o retorno foram os mesmos encontrados nas partes (d) e (e) dos problemas básicos que podem ser visualizados nas figuras 20 e 28 respectivamente. Os resultados desta parte também seguem dispostos nas figuras 30 e 31.

```
Command Window
New to MATLAB? See resources for Getting Started.
>> decdtmf(x1)

digit =
     4

digit =
     4     9

digit =
     4     9     1

digit =
     4     9     1     5

digit =
     4     9     1     5     8

digit =
     4     9     1     5     8     7
```

Figura 30: Decodificação DTMF de x1

```
Command Window
New to MATLAB? See resources for Getting Started.
>> decdtmf(x2)

digit =
     2

digit =
     2     5

digit =
     2     5     3

digit =
     2     5     3     1

j =
     0

digit =
     2     5     3     1     0
```

Figura 31: Decodificação DTMF de x2

6 Conclusão

Na primeira parte, implementamos o tom correspondente a cada dígito e a partir de sua TFTD usando a função *fft* do **MATLAB** para verificar se os sinais tem as frequências de pico correspondentes as apresentadas na Figura 1, usamos também a função *sound* para escutar se o som correspondia a tecla determinada. Posteriormente, criamos um vetor contendo vários dígitos espaçados e escutamos seu som.

Na segunda parte, decodificamos os sinais *x1* e *x2* presentes no arquivo **touch.mat**. Separando os dígitos dos sinais e comparando a *fft* destes com a *fft* esperada conseguimos decodificar os 2 números presentes nestes, 4915877 para *x1* e 2531000 para *x2*.

Na terceira parte, por meio da *fft* do sinal obtivemos os índices *k* correspondentes a cada frequência de pico do sinal e a partir da energia do sinal conseguimos observar que nestas está concentrada a maior parte da energia do sinal com essas informações implementamos a função *decdtmf* que decodifica automaticamente uma sequência presente em um sinal no formato de *x1* e *x2*. Para implementar a função separamos os dígitos e checamos se no índices correspondentes a cada pico se encontrava, na energia do sinal, um pico, assim sendo as duas frequências dos picos de energia do dígito o caracterizavam.

O projeto evidenciou bem como, usando dois sinais diferentes, se constrói o arranjo usado nas teclas de telefone e a importância da *fft* em sua implementação, além de como recuperar o sinal representado pelo sistema.

7 Apendice

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problemas B sicos %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2
3 %1)
4
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Digitos 0, 1, ... , 8, 9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6
7
8 d0=sin([0:999]*0.7217)+sin([0:999]*1.0247);
9 d1=sin([0:999]*0.5346)+sin([0:999]*0.9273);
10 d2=sin([0:999]*0.5346)+sin([0:999]*1.0247);
11 d3=sin([0:999]*0.5346)+sin([0:999]*1.1328);
12 d4=sin([0:999]*0.5906)+sin([0:999]*0.9273);
13 d5=sin([0:999]*0.5906)+sin([0:999]*1.0247);
14 d6=sin([0:999]*0.5906)+sin([0:999]*1.1328);
15 d7=sin([0:999]*0.6535)+sin([0:999]*0.9273);
16 d8=sin([0:999]*0.6535)+sin([0:999]*1.0247);
17 d9=sin([0:999]*0.6535)+sin([0:999]*1.1328);
18
19
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 1- (a) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21
22
23 d1=sin([0:999]*0.5346)+sin([0:999]*0.9273);
24 d2=sin([0:999]*0.5346)+sin([0:999]*1.0247);
25 d3=sin([0:999]*0.5346)+sin([0:999]*1.1328);
26 d4=sin([0:999]*0.5906)+sin([0:999]*0.9273);
27 d5=sin([0:999]*0.5906)+sin([0:999]*1.0247);
28 d6=sin([0:999]*0.5906)+sin([0:999]*1.1328);
29 d7=sin([0:999]*0.6535)+sin([0:999]*0.9273);
30 d8=sin([0:999]*0.6535)+sin([0:999]*1.0247);
31 d9=sin([0:999]*0.6535)+sin([0:999]*1.1328);
32 d0=sin([0:999]*0.7217)+sin([0:999]*1.0247);
33
34 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 1- (b) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
35
36
37 om=2*pi*[0:2047]/2048;
38 fftd2=fft(d2,2048);
39 fftd9=fft(d9,2048);
40 figure(1);
41 plot(om, fftd2.*conj(fftd2), om,fftd9.*conj(fftd9));
42 legend('dft d2', 'dft d9');
43 xlim([0 pi/2]);
44
45 %1c)
46 fonem = [d9 [0:100]*0 d9 [0:100]*0 d9 [0:100]*0 d1 [0:100]*0 d8
47          [0:100]*0 d1 [0:100]*0 d5 [0:100]*0 d8 [0:100]*0 d7];
48 fone = [d3 [0:100]*0 d2 [0:100]*0 d7 [0:100]*0 d1 [0:100]*0 d8
49         [0:100]*0 d2 [0:100]*0 d1 [0:100]*0 d0];
50 sound(fone);
51
52 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problemas Intermedi rios %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
53
54 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 2- (d) e (e) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
55
56 %separando x1
57 size(x1);
58 x11=x1(:,1:1000);
59 x12=x1(:,1101:2100);
60 x13=x1(:,2201:3200);
```

```

59 x14=x1(:,3301:4300);
60 x15=x1(:,4401:5400);
61 x16=x1(:,5501:6500);
62 x17=x1(:,6601:7600);
63
64
65 %separando x2
66
67
68 size(x2);
69 x21=x2(:,1:1000);
70 x22=x2(:,1101:2100);
71 x23=x2(:,2201:3200);
72 x24=x2(:,3301:4300);
73 x25=x2(:,4401:5400);
74 x26=x2(:,5501:6500);
75 x27=x2(:,6601:7600);
76
77
78 %fftx1
79
80
81 fftx11=fft(x11,2048);
82 fftx12=fft(x12,2048);
83 fftx13=fft(x13,2048);
84 fftx14=fft(x14,2048);
85 fftx15=fft(x15,2048);
86 fftx16=fft(x16,2048);
87 fftx17=fft(x17,2048);
88
89
90 %fftx2
91
92
93 fftx21=fft(x21,2048);
94 fftx22=fft(x22,2048);
95 fftx23=fft(x23,2048);
96 fftx24=fft(x24,2048);
97 fftx25=fft(x25,2048);
98 fftx26=fft(x26,2048);
99 fftx27=fft(x27,2048);
100
101
102 %fft digitos
103
104
105 fftx1=fft(x1);
106 fftd1=fft(d1, 2048);
107 fftd2=fft(d2, 2048);
108 fftd3=fft(d3, 2048);
109 fftd4=fft(d4, 2048);
110 fftd5=fft(d5, 2048);
111 fftd6=fft(d6, 2048);
112 fftd7=fft(d7, 2048);
113 fftd8=fft(d8, 2048);
114 fftd9=fft(d9, 2048);
115 fftd0=fft(d0, 2048);
116
117
118 %%%%%%%%% Sinais %%%%%%%%%
119
120

```

```

121 om2=2*pi*[0:7599]/7599;
122 figure(2);
123 plot(om2,fftx1.*conj(fftx1), om, fftd4.*conj(fftd4));
124 legend('dft sinal x1', 'dft d4');
125
126 figure(3);
127 plot(om,fftx11.*conj(fftx11), om, fftd4.*conj(fftd4));
128 legend('dft sinal x1 digito 1', 'dft d4');
129
130 figure(4);
131 plot(om,fftx12.*conj(fftx12), om, fftd9.*conj(fftd9));
132 legend('dft sinal x1 digito 2', 'dft d9');
133
134
135 figure(5);
136 plot(om,fftx13.*conj(fftx13), om, fftd1.*conj(fftd1));
137 legend('dft sinal x1 digito 3', 'dft d1');
138
139 figure(6);
140 plot(om,fftx14.*conj(fftx14), om, fftd5.*conj(fftd5));
141 legend('dft sinal x1 digito 4', 'dft d5');
142
143 figure(7);
144 plot(om,fftx15.*conj(fftx15), om, fftd8.*conj(fftd8));
145 legend('dft sinal x1 digito 5', 'dft d8');
146
147 figure(8);
148 plot(om,fftx16.*conj(fftx16), om, fftd7.*conj(fftd7));
149 legend('dft sinal x1 digito 6', 'dft d7');
150
151 figure(9);
152 plot(om,fftx16.*conj(fftx17), om, fftd7.*conj(fftd7));
153 legend('dft sinal x1 digito 7', 'dft d7');
154
155
156 figure(10);
157 plot(om2,fftx1.*conj(fftx1));
158 legend('dft sinal x2');
159
160 figure(11);
161 plot(om,fftx21.*conj(fftx21), om, fftd2.*conj(fftd2));
162 legend('dft sinal x2 digito 1', 'dft d2');
163
164 figure(12);
165 plot(om,fftx22.*conj(fftx22), om, fftd5.*conj(fftd5));
166 legend('dft sinal x2 digito 2', 'dft d5');
167
168
169 figure(13);
170 plot(om,fftx23.*conj(fftx23), om, fftd3.*conj(fftd3));
171 legend('dft sinal x2 digito 3', 'dft d3');
172
173 figure(14);
174 plot(om,fftx24.*conj(fftx24), om, fftd1.*conj(fftd1));
175 legend('dft sinal x2 digito 4', 'dft d1');
176
177 figure(15);
178 plot(om,fftx25.*conj(fftx25), om, fftd0.*conj(fftd0));
179 legend('dft sinal x2 digito 5', 'dft d0');
180
181 figure(16);
182 plot(om,fftx26.*conj(fftx26), om, fftd0.*conj(fftd0));

```

```

183 legend('dft sinal x2 digito 6', 'dft d0');
184
185 figure(17);
186 plot(om,fftx27.*conj(fftx27), om, fftd0.*conj(fftd0));
187 legend('dft sinal x2 digito 7', 'dft d0');
188
189 figure(18);
190 plot(om, fftd1.*conj(fftd1));
191 legend('dft d1');
192 xlim([0 pi/2]);
193
194 figure(19);
195 plot(om, fftd3.*conj(fftd3));
196 legend('dft d3');
197 xlim([0 pi/2]);
198
199 figure(20);
200 plot(om, fftd4.*conj(fftd4));
201 legend('dft d4');
202 xlim([0 pi/2]);
203
204 figure(21);
205 plot(om, fftd5.*conj(fftd5));
206 legend('dft d5');
207 xlim([0 pi/2]);
208
209 figure(22);
210 plot(om, fftd6.*conj(fftd6));
211 legend('dft d6');
212 xlim([0 pi/2]);
213
214 figure(23);
215 plot(om, fftd7.*conj(fftd7));
216 legend('dft d7');
217 xlim([0 pi/2]);
218
219 figure(24);
220 plot(om, fftd8.*conj(fftd8));
221 legend('dft d8');
222 xlim([0 pi/2]);
223
224 figure(25);
225 plot(om, fftd0.*conj(fftd0));
226 legend('dft d0');
227 xlim([0 pi/2]);
228
229 figure(26);
230 plot(om, fftd9.*conj(fftd9));
231 legend('dft d9');
232 xlim([0 pi/2]);
233
234 figure(27);
235 plot(om, fftd2.*conj(fftd2));
236 legend('dft d2');
237 xlim([0 pi/2]);
238
239 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Problemas Avancados %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
240
241
242 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 2 - (f) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
243
244 k1=[ceil(0.5346*2048/(2*pi)) ceil(2048*0.9273/(2*pi))];

```

```

245 k2=[ceil(0.5346*2048/(2*pi)) ceil(2048*1.0247/(2*pi))];
246 k3=[ceil(0.5346*2048/(2*pi)) ceil(2048*1.1328/(2*pi))];
247 k4=[ceil(0.5906*2048/(2*pi)) ceil(2048*0.9273/(2*pi))];
248 k5=[ceil(0.5906*2048/(2*pi)) ceil(2048*1.0247/(2*pi))];
249 k6=[ceil(0.5906*2048/(2*pi)) ceil(2048*1.1328/(2*pi))];
250 k7=[ceil(0.6535*2048/(2*pi)) ceil(2048*0.9273/(2*pi))];
251 k8=[ceil(0.6535*2048/(2*pi)) ceil(2048*1.0247/(2*pi))];
252 k9=[ceil(0.6535*2048/(2*pi)) ceil(2048*1.1328/(2*pi))];
253 k0=[ceil(0.7217*2048/(2*pi)) ceil(2048*1.0247/(2*pi))];
254 kn=[k1 k2 k3 k4 k5 k6 k7 k8 k9 k0];
255 w=[0.5346 0.9273 1.0247 1.1328 0.5906 0.6535 0.7217];
256
257 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% 2 - (g) %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
258
259 figure(1);
260 hold on;
261 fftd8=fft(d8, 2048);
262 p=fftd8.*conj(fftd8);
263 om=2*pi*[0:2047]/2048;
264 plot(om,p);
265 plot([w(1),w(1)],[0,270000]);
266 plot([w(2),w(2)],[0,270000]);
267 plot([w(3),w(3)],[0,270000]);
268 plot([w(4),w(4)],[0,270000]);
269 plot([w(5),w(5)],[0,270000]);
270 plot([w(6),w(6)],[0,270000]);
271 plot([w(7),w(7)],[0,270000]);
272 xlim([0 pi/2]);
273 legend('energia dft digito 8', 'w = 0.5346', 'w = 0.5906', 'w =
    0.6535', 'w = 0.7217', 'w = 0.9273', 'w = 1.0247', 'w =
    1.1328');
274 hold off;
275
276 function digit = decdtmf(x)
277     s=(size(x)+100)/11;
278     digit = [] ;
279
280     for i = 0:s
281         init = 1100*i+1;
282         d = X(init:init+999);
283         for j = 1:size(kn)
284             if d(kn(j,1)) > 500 && d(kn(j,2)) > 500
285                 digit(end+1)= j
286             end
287         end
288     end
289 end
290 end

```

Bibliografia

- [1] - Alan V. Oppenheim, Ronald W. Schaffer. Discrete-Time Signal Processing, 3rd Edition
- [2] - Richard Blahut . Fast Algorithms for Signal Processing
- [3] - Ricardo Menezes Campello de Souza. Processamento Digital de Sinais, Projeto III (2020.3) - Detecção de Sinais DTMF
- [4] - Juliano Bandeira Lima. Decodificação de Sinais DTMF via transformada Aritmética de Fourier, 2004.

https://www.ufpe.br/documents/39830/1359036/108_julianoLima/25899ad5-c61b-4393-9329-f9cc0f64f399